SCIENCE & TECHNOLOGY

A review of coding theory

by Brian P. McArdle

1. Introduction

The general area of Coding Theory for telecommunications and computer applications is reviewed to provide a simple introduction to the subject. For further information, the reader should consult the books in the reference section.

There is no formal definition of a code. Essentially, messages are represented in some form more easily transmitted than normal written language. In this article, a code is a digital electronic signal that represents a message symbol, such as a letter or number. For example, a teleprinter code would have to have a signal for every possible symbol (26 letters, 10 numerals and other symbols) and signals for every operation (that is, space, carriage return and line feed controls). Figure 1 shows the arrangement.

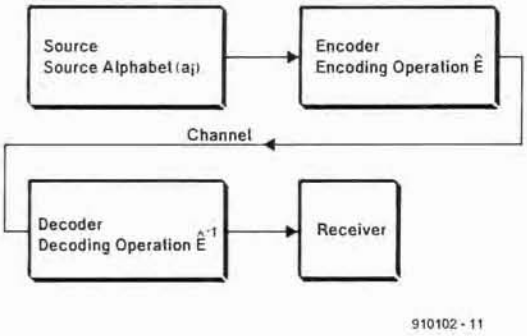


Fig. 1. Encoding/decoding operation.

An encoding operation Ê turns a message symbol a_j into coded form for transmission over a channel. The set $\{a_j\}$ is the source alphabet and $\{\Pr(a_j)\}$ is the set of probabilities associated with this alphabet: $\Pr(a_j)$ is the probability that a_j occurs. In normal language, this is the probability of occurrence of letters. The word 'channel' has a general meaning. It could be a cable, radio link or storage medium where the receiver is retrieving the messages at some later time. Obviously, the receiver must be able to apply a decoding operation $\hat{E}^{-1}$. Hence, the principal requirement for a satisfactory code is that the coded symbols be uniquely decodable. In mathematical terms, $\hat{E}[a_j]$ cannot represent more than one symbol. $\hat{E}[a_i]$ cannot equal $\hat{E}[a_j]$ unless a_i and a_j are effectively the same symbol. For example, Ê might not distinguish between upper and lower case letters. The decoded messages may be printed in upper case letters only. Thus, apart from small variations that should not affect normal understanding, the encoding operation, irrespective of its complexity or purpose, must be exactly and uniquely reversible.

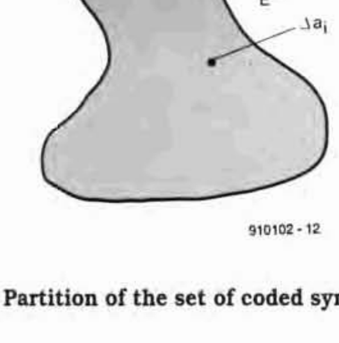


Fig. 2. Partition of the set of coded symbols.

A more formal mathematical definition is that the set of coded symbols $\{\hat{E}[a_j]\}$ must be uniquely partitioned (that is, can be divided into subsets that do not overlap) such that each partition can be associated uniquely with a source symbol. Figure 2 shows the arrangement.

The remainder of this article attempts to explain the meaning of Ê in different applications, such as error-detection-correction and encryption. It is always assumed that the encoding operation is uniquely reversible unless otherwise stated. Another important assumption is that a *memory-less source* is involved. The probabilities $\Pr(a_j)$s are the probabilities of the general occurrence of these symbols in normal language. In reality, letters occur in groups (digraphs and trigraphs). *i* before *e*, except before *c* is a well-known expression. Consequently, the probability of occurrence of a particular letter could be influenced by preceding letters. It is also assumed that a memory-less source is being considered unless otherwise stated. The analysis is mostly confined to digital coding except for Section 6, which deals with coding for analogue signalling.

2. Different codes

Codes can be analysed from many different viewpoints, but engineers are generally concerned only with two main categories.

(a) Fixed length codes

Every character of such a code is represented by a block of bits with every block having the same length. A typical example would be a computer code, such as ASCII or EBCDIC. Both of these use blocks of eight bits. Thus, there is a total of 2^8 or 256 difference blocks. Any two blocks of the same code would have to differ in at least one bit. The blocks need not be symbols (letters, numbers, punctuation marks) but can be controls (carriage return, line feed, etc.).

(b) Variable length codes

Consider an alphabet of five symbols $\{a, b, c, d, e\}$. In **(a)**, this would require a code of three bits per block or symbol with $2^3 - 5 = 3$ redundant blocks. However, if the following arrangement of three blocks of two bits per block and two blocks of three bits per block is used,

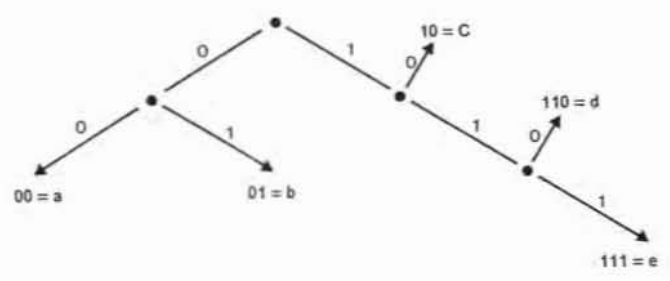
$$a = 00,\ b = 01,\ c = 10,\ d = 110,\ e = 111,$$

the average length of a message would be reduced. Blocks still have a specific length, but it is no longer the same fixed length. The basic requirement for unique decoding must still be maintained. For example, the bit sequence **011100011110** is easily decoded to ***bdaec*** with no errors. This must apply for all combinations of the symbols. An important quantity is the average length *L* of a block given by

$$L = \sum_j p_j \tau_j \qquad \text{[Eq. 1]}$$

where p_j is the probability of occurrence of block type j with τ_j bits. Ideally, this should be as small as possible to minimize the total number of bits per message. To ensure this requirement, the large probabilities would be paired with the smaller blocks. Morse code is another example where the common letter *e* is a dot, but *z* is two dashes followed by two dots.

This particular example has a special significance in addition to variable length blocks. If it is rewritten in the form of a diagram as below, it seems to have a tree-type structure with different branches.



Each branch is terminated by a symbol. The branches join together at nodal points which do not in themselves represent symbols. This arrangement indicates an *instantaneous* code. This means that the decoding operation does not require a 'memory', that is, it does not refer to blocks before or after any block that is being decoded. In the decoding of ***bdaec***, it was not necessary to test the 3rd bit before deciding that the first two bits, 01, represented *b*. This property remained true for the full operation and for all decoding operations irrespective of the combinations of symbols. (This should not be confused with a memory-less source, defined earlier, where there is no relationship between the occurrence of different symbols.). In an instantaneous code, no block can be a prefix or suffix for another block. Huffman codes, which are too involved to be considered in a simple overview, come into this category. However, it must be emphasized that any collection of blocks of varying lengths does not make an instantaneous code. There is a specific requirement given by the *Kraft inequality*

$$\sum_j \left(\frac{1}{2^{\tau_j}}\right) \le 1$$

to form such a code. Further analysis is outside the scope of this paper and the reader is referred to the Reference Section for further study.

3. Information theory

Information theory has steadily increased its profile over the past few years and it is no longer possible to study telecommunications, especially coding, without touching on it somewhere. At first glance, the ideas behind it can appear too general and abstract for simple, direct applications. The fundamental fact is that the basic concepts of entropy, equivocation and channel capacity come from information theory, which, in turn, has influenced coding theory, and require some explanation.

There is a fundamental difference between an electronic signal and its value as information. In sound broadcasting, un unmodulated carrier would not convey any programme content to a listener. Therefore, there is a need to be able to quantify the value of a signal as information. In the 1920s, Hartley put forward the idea that the *logarithmic function* could be used as a measure of information. This was one of the landmarks in information theory. If two messages, a_i and a_j, are independent,

$$\log\{\Pr(a_i) \text{ and } \Pr(a_j)\}=\log\{\Pr(a_i)\}+\log\{\Pr(a_j)\} \qquad \text{[Eq. 2]}$$

and the base 2 is normally used. Remember that 'log' is not a linear function. The idea that the information contents of two independent messages is simply the sum of the information of each separate message seems instinctively correct. However, this method of measuring information has no connection with an actual signalling system. The *entropy* for $A=\{a_j\}$ is given by

$$H(A) = -\sum_j \Pr(a_j)\log\Pr(a_j) \qquad \text{[Eq. 3]}$$

and is a measure of the average information. An alternative explanation, which has become more common in recent years, is that it is a measure of the uncertainty in the information. $H(A)=0$ means that $\Pr(a_j)=1$ and $\Pr(a_i)=0$ for all other messages. Consequently, there is no doubt about the message. The maximum value occurs when the probabilities are the same and all messages are equally likely. If there are *n* possible messages, H(*A*) is between 0 and log(*n*). The dimension is *information bits per symbol*.

To apply information theory to coding theory, consider Fig. 3 where there is a noisy channel between sender A and receiver B. The joint entropy is given by the equation

$$H(A,B)=H(A)+H(B/A) \qquad \text{[Eq. 4]}$$

where H(*B*/*A*) is known as the conditional entropy or *equivocation*. This in turn is defined as

$$H(B/A) = \sum_j H(B/a_j)\Pr(a_j). \qquad \text{[Eq. 5]}$$

In non-mathematical terms, H(*B*/*A*) is a measure of the information loss in transmission. The *channel capacity* is given by

$$C(A,B)=\text{maximum } H(A)-H(A/B). \qquad \text{[Eq. 6]}$$

This appears correct because the limit on the information conveyed over a channel is determined by the original uncertainty of that information (before reception) reduced by the uncertainty after reception. Essential capacity is limited only by noise and the *Hartley-Shannon law* sets an upper limit of $W\log(1+S/N)$, where *W* is the information bandwidth, *S* is the signal power and *N* is the noise power. For technical reasons, present-day systems operate well below this limit. The reader is referred to the Reference Section for further study.

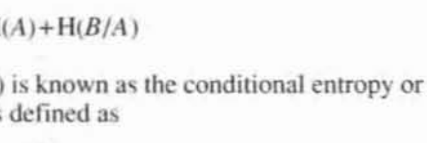


Fig. 3. Communications channel.

From the point of coding and electronic engineering, Eq. 6 can be simplified for normal use. Consider the case for a binary channel where *A* and *B* represent the input and output respectively. In general, the probability of a '0' or '1' is $^1/_2$, which gives $H(A)=\log(2)=1$. (The entropy of the source alphabet could be computed from the probability of occurrence of the various symbols, but it is the channel that is now under consideration.). If *p* is the probability of an error where a '0' is received as a '1', or vice versa, the channel conditional probabilities are

A	B	
	0	1
0	$(1-p)$	p
1	p	$(1-p)$

From Eq. 5:

$$H(A,B)=\Pr(0)[-(1-p)\log(1-p)-p\log(p)]+ \Pr(1)[-(1-p)\log(1-p)-p\log(p)] \qquad \text{[Eq. 7]}$$

which gives a new expression for the *channel capacity*:

$$C(A,B)=1+p\log(p)+(1-p)\log(1-p) \qquad \text{[Eq. 8]}$$

in *bits per symbol*. This is the usual expression in most textbooks on telecommunications. If the signalling rate is R symbols per second, the right-hand side is multiplied by R to give bits per second. Thus, information theory can be useful in the analysis of codes. The entire area has become extensive and has been treated only superficially here.

4. Error detection and correction

Error detection and correction is one of the main applications of coding theory and paralleled its development. Figure 3 showed the problems with errors where a '0' can be received as a '1' or vice versa. The use of the word 'receiver' is general in that it could represent a storage medium, and so on. It suffices to say that data is corrupted, which limits its value upon reproduction. Section 3 demonstrated that channel capacity is limited only by noise. To reduce the effects of errors, and therefore noise, extra bits are added to a block of data bits to create new and larger blocks, which in turn allow errors to be identified.

Consider the (7,4) *Hamming code* as follows:

Position:	7	6	5	4	3	2	1	$c_4=(d_7+d_6+d_5) \bmod 2$
Bit:	d_7	d_6	d_5	c_4	d_3	c_2	c_1	$c_2=(d_7+d_6+d_3) \bmod 2$
								$c_1=(d_7+d_5+d_3) \bmod 2$

There are three check bits in positions 1, 2 and 4 which have been derived from the data bits in the other four positions. The code is *linear* in the sense that the check bits are linear combinations of the data bits and the encoding operation is simply the application of the three linear equations. Since every block will have a total of seven bits without exception, the code is in the fixed length category. The position of the check bits within the block is very significant. A receiver generates the check bits from the received data bits and applies the decoding rule in Appendix 1. For example, if d_3 has been altered, c_1 and c_2 will not be validated and so on. The arrangement to check five data bits is given in Appendix 2. In both these examples, the set of coded blocks is such that the minimum variation between any two blocks in the same set is three bits. This is known as the *Hamming distance*. The reader is referred to Reference 2 for a more detailed explanation. The main point to note is that the method identifies only one error per block. In general, r check bits have (2^r-1) possible combinations and thus r bits in a total size of n bits must satisfy the condition $(2^r-1)\geq n$ in order to identify and therefore correct one error. To correct two or more errors per block, a code with a larger Hamming distance and more complicated arrangement would be needed.

Cyclic codes are the most commonly used for error detection and correction. These are also of the fixed length variety. For a block of total size n, the check bits are produced by a generator polynomial which is a factor of (x^n+1). A typical example is the specification MPT 1317 for the transmission of data over radio links. The format is as follows:

DATA	CHECK	PARITY
Bit 64 63 62 17	16 15....2	1

with a block size of 64 bits. However, the first bit is for parity and is generated by the other 63 bits. The 15 check bits are generated from the 48 data bits using the generator polynomial

$$g(x)=x^{15}+x^{14}+x^{13}+x^{11}+x^4+x^2+1 \qquad \text{[Eq. 9]}$$

which is a factor of $(x^{63}+1)$. Refer to Appendix 3 for an exact breakdown. The data bits are the coefficients of the terms x^{62} down to x^{15} inclusive. Some books write the data bits on the right-hand side of the format, but this is not important provided they represent the high power terms of the polynomial. The polynomial consisting of only the data bits is divided by $g(x)$. The remainder is then added back to produce a new polynomial such that $g(x)$ is now a factor of the new polynomial. Since the check bits are essentially the original remainder, they represent the terms x^{14} down to x^0. Then a parity bit is added in order to detect odd numbers of 1s and the full 64-bit block should have even parity. Refer to Appendix 4 for the generation of a parity bit. The overall result is that the code can identify and correct up to four errors per block. This is a considerable improvement on the (7,4) Hamming code, but the operation is much more involved and the block size nine times larger. To check for errors, the receiver divides the polynomial by $g(x)$ and there should be no remainder..

Another example is the *POCSAG* code for paging, which uses the format

DATA	CHECK	PARITY
Bit 32 31 30....12	11....2	1

and the generator polynomial

$$g(x)=x^{10}+x^9+x^8+x^6+x^5+x^3+1 \qquad \text{[Eq. 10]}$$

which is a factor of $(x^{31}+1)$. Refer to Appendix 5 for an exact breakdown. The overall method is the same with the 21 data bits generating the 10 check bits to produce a 31-bit block plus an extra parity bit.

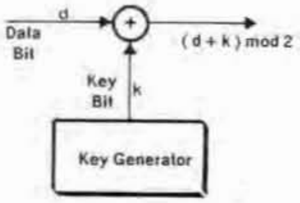
5. Encryption

In encryption, the $\hat{E}$ operation, defined in Section 1, represents a secrecy operation and is usually written as $\hat{E}_K$ in most textbooks. The parameter K is known as the key and its purpose is to vary the operation. This is in complete contrast to error-detection-correction where exactly the same operation is performed on all blocks without exception. The importance of K is that it is generally the part of $\hat{E}_K$ that is kept secret. In a publicly known algorithm, such as the *Data Encryption Standard* (DES), the complete algorithm is known. A user chooses a key from the set of possible keys $\{K\}$ and encrypts the data. Thus, only encrypted data appears on the channel of Fig. 1. The data can be recovered by the inverse or decryption operation $\hat{E}_K^{-1}$ which also requires the correct key. If the key in use is kept secret and only known to authorized receivers, the data is kept secret. Obviously, $\{K\}$ must be sufficiently large to prevent an unauthorized user from trying each key in turn. There are a number of other requirements that are outside the scope of this paper.

There are three main methods of encryption.

(a) Stream encryption

In Fig. 4, each bit of the data is added modulo 2 using an XOR logic operation. A sequence of key bits is produced by the key generator such that each data bit is encrypted by its own particular key bit.



910102 - 14

Fig. 4. Stream encryption.

The authorized receiver must know the method of key generation to reproduce the exact same sequence. The inverse operation is simply to apply the key sequence in the correct order to the sequence of encrypted bits. It would be too complicated to discuss the various techniques of key generation, but the most common method uses shift registers to generate a pseudo-random binary sequence. Generally, part of this process must be kept secret, such as the number of stages and feedback arrangements. The current proposals to provide encryption facilities on the cellular system GSM or for digital short-range radio DSRR are believed to use a form of stream encryption. However, the information is confidential and it is very likely that the exact method will not be made public.

(b) Block encryption

Block encryption—see Fig. 5—differs from stream encryption in that a block is encrypted as a single unit. The most widely known method is the US *Data Encryption Standard*, which uses blocks of 64 bits for input and output. The algorithm was published in 1977 and the exact method is for public information. The actual key is a 56-bit block, so that the number of possible keys is 2^{56}. In operation, the authorized receiver would know the particular key in use and apply the inverse or decryption algorithm. Controversy has always surrounded the key size and recent articles have suggested methods for an improved DES.

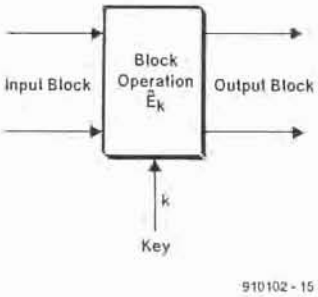


Fig. 5. Block encryption.

The main advantage over (a) is that a satisfactory block operation creates interdependence between the bits of a block. If one bit of an input block is varied, a number of bits in the output block are altered. However, it is generally much slower than stream encryption and cannot be used for high-speed telecommunications applications.

(c) Public key encryption

Public key encryption differs from the methods in (a) and (b) in that part of the key is made public, whence its name. The main requirement is that the part which must remain secret should not be easily deducible from the public part. A typical example is the RSA method introduced in 1978. Each user publishes two numbers, N and e. N is very large, of the order of 80 digits, and the product of two primes, P and Q, while e and d satisfy the equation

$$1 = ed \bmod (P-1)(Q-1). \qquad \text{[Eq. 11]}$$

Only e is made public; d, P and Q remain secret. If user A wishes to forward the message 'a' to user B, A looks up the parameters N and e for B and transmits

$$b = a^e \bmod N. \qquad \text{[Eq. 12]}$$

User B recovers the original message from

$$a = b^d \bmod N; \qquad \text{[Eq. 13]}$$

since d is one of the secret parameters, this cannot be done by any other user. From a secrecy point of view, an unauthorized user would have to factor N into P and Q to calculate d. Thus, as long as N is sufficiently large, this is impractical and the method is secure. There are other methods, such as the *Merkle-Hellman-Knapsack Method*, but they all follow the same principle of a public and a private key. Equations 12 and 13 are the equivalents of the encoding and decoding operations.

6. Coding for analogue signalling

In the preceding four sections, it was assumed that digital signal processing was in use. However, codes are also used in analogue electronics, but their application is rather limited. For example, in the PMR service, CTCSS (continuous tone controlled signalling system) has been around for a number of years. During transmission, an encoder generates a specific audio tone that modulates the radio frequency carrier. This tone is continuous for the duration of a message. In the absence of a CTCSS signal, the decoder at the receiving end is deactuated.

Another example is tone selective calling, such as EEA and ZVEI. In these methods, a sequence of five tones is used to form an address for a receiver. Both EEA and ZVEI have a total of 12 possible tones. Each possible address consists of a set of five, which actuates the receiver from the point of the user. However, despite these examples, coding has remained almost exclusively digital and the cellular GSM standard actually prohibits the use of tones.

On the secrecy side, there are voice scramblers that use frequency inversion, but increasingly the trend has been to digitize speech (for instance, ADPCM—Appendix 6) and to apply the techniques of Section 5. Coding in analogue signal processing is very restricted and need not be considered seriously.

7. References

(1) *Information Theory and its Engineering Applications*
D.A. Bell; Pitman (1968)

(2) *Coding and Information Theory*
Richard W. Hamming; Prentice-Hall (1980)

(3) *Information and Coding*
J.A. Llewllyn; Chartwell-Bratt (1987)

(4) *A First Course in Coding Theory*
Raymond Hill; Oxford University Press (1986)

(5) *Codes and Cryptography*
Dominic Welsh; Oxford University Press (1988)

(6) *Coding for Digital Recording*
John Watkinson; Focal Press (1990)

Appendix 1

The receiver re-calculates the check bits and validates them against the received values.

c_1 and c_2 are not validated ⇒ d_3 is incorrect
c_1 and c_4 are not validated ⇒ d_5 is incorrect
c_2 and c_4 are not validated ⇒ d_6 is incorrect
c_1, c_2 and c_4 are not validated ⇒ d_7 is incorrect

The sum of the indices of the check bits indicate the location of the erroneous bit. The correction process replaces a '0' by a '1' or vice versa. The principal difficulty is that two errors can cause a correct bit to be changed.

Appendix 2

The Hamming code for five data bits is $d_9\, c_8\, d_7\, d_6\, d_5\, c_4\, d_3\, c_2\, c_1$ and requires four check bits with the same procedure as in the (7,4) code. r check bits can test up to 2^r-1 locations. For r=3, this gives $n=2^3-1$, which leaves four data bits. For r=4, there is a block size n=15 which allows for 11 data bits and four check bits in the order:

$$d_{15}\ d_{14}\ d_{13}\ d_{12}\ d_{11}\ d_{10}\ d_9\ c_8\ d_7\ d_6\ d_5\ c_4\ d_3\ c_2\ c_1.$$

Appendix 3

For a 63-bit block, the factors of the modulus are: